



Daniel Bunte

[Follow](#)

Leader, self-taught developer with interests in AI/FBP/Gaming/Automation/Trading.
Feb 22 · 3 min read

Why I'm still using "Fizz Buzz" to hire Software-Developers



Hiring is a long and sometimes tedious process. As a hiring manager you want to make sure you filter people out in early phases.

How can you do this? Current measures include, but are not limited to phone screening, personal interviews, or technical tests. In my opinion, all of these are very important, since your company is interested in a lot of aspects in a person. For instance, communication skills are becoming more important, even for software developers, because processes change and include more and more teamwork.

But in this article, I'll be focussing on hard technical programming and logic skills. There's been a simple, but effective test in the industry that's been used by hundreds of companies for decades (I think). It's called "Fizz Buzz" and the problem to solve is rather simple.

Write a program that prints the numbers from 1 to 100. If it's a multiple of 3, it should print "Fizz". If it's a multiple of 5, it should print "Buzz". If it's

| a multiple of 3 and 5, it should print "Fizz Buzz".

Most people start writing some code immediately, because they recognize the pattern of a **for-loop**. Oh, that's great, they know a little syntax!

But software development has never been about **writing code**. The essence is always to solve a problem—that's it. Writing code is just one of many tools used to achieve this goal. Think about GUI-Design. Is it part of software development? Definitely! Does it include writing code? Well, sometimes.

So what do I want to see a possible candidate doing? I want to see them thinking, maybe even discussing a solution and then start with a simple mockup. At some point they will discover to check for modulo. Later, they'll see checking modulo of 3 doesn't make sense as first instruction in the loop. And maybe they'll even discover to combine 3 and 5 in one check.

What I've seen is interesting at least. People didn't know the modulo operator, so they wrote "modulo"—OK. People didn't know about modulo at all, so they used `Math.floor(number/3)`—unexpected, creative, but a valid solution. People tried to be smart concatenating strings, but missing the space in-between—bad. People used nested ifs, ending in code not working, while claiming themselves senior with 15 years experience—very bad.

Don't get me wrong; candidates won't fail the job interview just because of this. As I mentioned before, there's a lot more you want to know about and there are a lot more things people can impress you with.

I still think this simple exercise is still a quick way to get a first impression.



